

Visual Control of a Remote Vehicle

David Sanchez-Benitez¹, Jesus M. de la Cruz¹,
Gonzalo Pajares¹, and Dawei Gu²

¹ Department of Computer Architecture and Automatics
Universidad Complutense de Madrid, Spain

{davisanc,jmcruz}@fis.ucm.es, pajares@dacya.ucm.es

² Engineering Department, Leicester University, UK
dag@le.ac.uk

Abstract. We present a method for locating and determining the six degrees of freedom through a simple algorithm based on artificial vision. This algorithm can estimate the relative orientation of the camera with respect to a precise figure, it gives, roll, pitch and yaw, as well as the distance to the figure. We make use of the Euler number of a set of figures with a given distribution. We use the system to drive a small radio controlled car with the only assistant of the information gathered by a standard web-cam.

Keywords: Artificial Vision, Computer Vision, Control, Autonomous Vehicles, six Degrees Of Freedom (6DOF).

1 Introduction

In recent years there has been a huge increase in the study and manufacturing of autonomous aerial vehicles [1][2]. Of special civil interest are those aerial vehicles with hovering capability, for their possible use in search, locating and rescue. Part of the project in which the ISCAR group, from UCM, is involved, is the co-operation of several aerial and marine autonomous vehicles for the use in rescue missions, minimizing the risks for human teams. Among the qualities these aerial vehicles should have, is the autonomous vertical take off and landing (VTOL) on the ships, even when these are not stabilized or halted. When an Unmanned Aerial Vehicle (UAV) has to land, it has to recognise the touch-down spot somehow. Thus, it is necessary that some kind of system based on artificial vision is present onboard to help in the manoeuvre. Starting with the detection of a two-dimensional figure, which is easily reproducible, a small commercial web-cam detects its state, that is, the respective angles of inclination the web-cam has with respect to the figure (roll, pitch and yaw) and the distance to this figure, enabling the projection of the spatial coordinates on the plane to locate the camera in the three-dimensional world. This extends the applicability of the system to the reader's imagination. There are several publications regarding the same problem but taking a different perspective to solve the problem ([5][6][3][7][9][10]). None of them determine the whole state of the camera, i.e. the vehicle carrying it, relative to the landing pad.

2 The Vision System

2.1 Why Euler?

To use the vision system as a locating system, the properties or characteristics that are going to be used to distinguish the target from the surroundings of the image have to be selected. From the beginning, it was decided not to use stereo vision (as done in [11]) to simplify the algorithm. We decided to test geometrical characteristics of some patterns, instead of colour analysis or the matching with a base figure, processes that are huge system-resource consumers, slowing down the process, and also noise sensitive being affected by the changing illumination conditions. In our case, we opted for distinguishing precise figures through their Euler number (number of connected regions or objects minus number of holes within the figures ([4])). The determination of this geometrical property is comfortably integrated within the package software we used for this study (MATLAB 2007b), its calculation requires very little computational time and allows us to distinguish and catalogue different figures in a picture rapidly. The implementation of the algorithm shown in [8] for calculating this property was considered first and later discarded for practical reasons, since the software mentioned above gave us more than sufficient processing speed.

2.2 The Figure and Its Raison d'être

Another factor to decide in our identification system is which pattern to use. As mentioned previously, the geometrical property used for identification is the Euler number. When we take and analyse images around us, randomly, it can be observed that regions with Euler numbers -2, -1, 0, 1 are normal, whereas, those with an Euler number smaller than -3 are quite unusual. For that reason, a pattern whose Euler number is -4 is chosen. This pattern is made out of three elements: the main element is a circle with five inner regions (holes) symmetrically distributed. The main property of this element is that its Euler number equals -4, which makes it easily distinguishable in the picture, since it is unusual for a geometrical entity to possess that property. The area of this part of the pattern is used in the calibration of the camera, for a latter determination of the distance from the camera to the target figure. The second element is situated inside one of the peripheral holes of the first element. It is an ellipse with two circular holes inside, thus it has an Euler number of -1. Finally, at the centre of the main element, a ring is included (with an Euler number of 0). The final pattern is represented in figure 1. For each element, besides the mentioned Euler number, the centre of mass (CM) is computed. The coordinates of the CMs of each one of the elements is the main information used for the calculation of the state variables. With these coordinates, we calculate the yaw angle as follows: Set an imaginary vertical line from top to bottom of the picture's plane, crossing the centre of the figure, given by the coordinates of the CMs of the main element, with Euler number equal to -4, which we will call CM_4 from now on for clarity. Next, draw the line connecting CM_4 with the CM of the ellipse (CM_1).

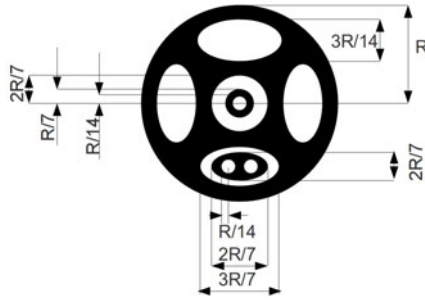


Fig. 1. Pattern used for detection and its proportions

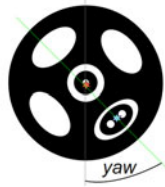


Fig. 2. Determination of the yaw angle

The angle formed by these two lines is the yaw angle of the camera with respect to the figure. (See figure 2) For the calculation of the pitch and roll we make use of the following: When a figure is observed it forms a certain angle with respect to its vertical, the figure appears distorted to the beholder, in a way such that the closest region will seem to have more mass than the farthest region (figure 3).

This distortion makes the CMs of the observed figure move closer to the position of the observer. Then, the greater the angle of observance, the greater the distortion and subsequently, the greater the displacement of the CMs. This way, since the images are captured under perspective projection, the main figure, when it is not observed exactly from its vertical, will present its CM displaced to the observer. The difference between the main region and the central ring is quite significant (in terms of area), hence, when the whole pattern is observed, the CM of the ring (CM_0) will barely alter its position, while the CM_4 will be seen clearly displaced from its centre, as much as the point of view. Thus, once we have the coordinates of both points, (CM_0 and CM_4) we will be able to obtain two distances: a vertical separation (what we call D_{ry} in the algorithm) and an horizontal one (D_{rx} , see figure 3). The horizontal distance between both CM of the main element and the small central ring will be proportional to the lateral inclination of the camera (roll) and the vertical distance between them will be proportional to pitch angle.



Fig. 3. D_{xy} and D_{xy} : the vertical and horizontal distance between CM₄ and CM₀

3 Algorithm

3.1 Relative Pose

Processing images is a very system-demanding task. It is important to proceed efficiently and to accelerate the process it is recommended to analyse the smallest portion of the image as possible. When the analysis starts, the whole image is analysed searching the figure in the largest possible region. If the target is found (region with Euler number -4), the region of interest (roi) is reduced to a region slightly bigger than the target itself. If the target is lost, the search restarts using the whole frame again. Once the target is located, the region of interest is extracted. This roi is centred on the target figure and wraps it so that in the next image the target will be inside this roi. This section of the picture that includes our target, is transformed into its complementary image—an image where the value of each pixel is 255 minus the value of the original pixel. The complementary image is then filtered and converted into a binary. The algorithm is similar to that used in [5]. The time invested by the system (CORE2 DUO T7700, 2GB RAM, MATLAB 2007b) was studied in taking and analysing one hundred images for different configurations in the sequence of the process. Table 1 synthesizes our results. Therefore, the first sequence of analysis, will provide an algorithm capable of taking and analysing 25 images per second. In the processed image, we will only have the characteristic figure of our platform (the pattern may be used as taking off/landing platform), which contain the three regions that interest us (figure 4). For another series of measurements, 519 images were taken from which the our pattern was detected in 492 ocasiones (95%). Keeping the relative pose of the camera (fixed angles and distance), the standard deviation for the yaw angle was 0.1 degrees, and for the distance to the pattern, 0.03 cm.

Table 1. Invested time for every sequence of analysis

Sequence	Invested Time/100 frames (s)
Image - roi - complementary - binary	3.9
Image - complementary - binary	6.3
Image - roi - binary -complementary	7.2
Image - binary - complementary	8.3

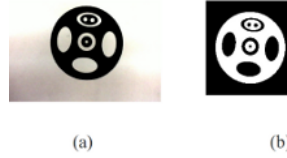


Fig. 4. a) Image taken by the camera and b) after being processed

3.2 Spatial Location

In order to locate the pattern spacially we use the position of the centre of mass (CM(1),CM(2)) of the main element of the pattern within the frame taken by the camera, whose origin lies on the upper left corner. We have to bear in mind that the algorithm analyses only a part of the frame taken by the camera, the region of interest, to which the position of this region of interest has to be added to the coordinates of the centre of mass in question. If the position of the region of interest inside the global image is given by the coordinates [lim1 lim2 lim3 lim4], where lim1 and lim2 indicate the position of the upper left corner of the region of interest in the global image, and lim3 and lim4 are, respectively, the height and width of this region of interest (see figure 5 for clarity). Then, we can assign the following values to the position of the figure: $X = CM(1) + lim1$ and $Y = CM(2) + lim2$. The drawback in the preceding method lies in the aberration introduced by the camera lens, which produces a distortion in the field of vision, rounding the edged of the image. This aberration might be ignored in cases where huge precision is not needed, as ours. Furthermore, for each variable, the rate of change is obtained from the time used in every iteration of capture-analysis and the previous measurement of the variable. With this, we are able to estimate the twelve state variables of the system under observance: three Euler angles, three spatial coordinates and all their rate of change.

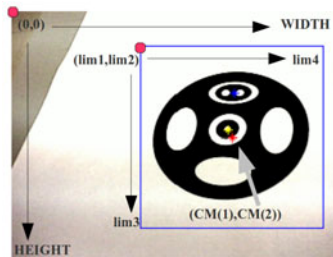


Fig. 5. Image of the platform showing the CMs

4 Constrains

Certain conditions are introduced in the detection process to insert robustness in the determination of the pattern. The first limitation introduced is the minimum area the pattern can present in the image taken by the camera. Experimentally,

it has been determined that if that area (number of pixels) is smaller than 800 (pixels), the different elements that make up the pattern can not be distinguished within it by the lost of sharpness, from which, an object whose Euler number is -4 but presents an area smaller than 800, is not considered as our pattern. This case is presented when the camera is too far from the pattern.

The second restriction imposed over the location process deals with the possible distance between CM_4 and CM_1 , the main element and the ellipse fitted in one of the ellipse-like holes. This distance must be smaller than the radius of the main element, otherwise, the element with Euler number -1 could not lie inside the pattern. The big issue here (not that big) is to obtain the area of the pattern. To do this, we can concentrate in figure 1 that presents the proportions of our pattern. Firstly, we consider the figure-pattern as a disk of radius R from which four ellipses and another smaller disk are extracted. The semimajor and semiminor axis of these ellipses are $3R/7$ and $3R/14$ respectively. These apparently strange proportions loose any strangeness and become whole numbers when R equals 7, the original radius used in the first test run with the algorithm. Following basic mathematics:

$$area_{pattern} = \pi R^2 - 4\pi \frac{3R}{7} \frac{3R}{14} - \pi \left(\frac{2R}{7}\right)^2. \quad (1)$$

from which the following constrain is obtained:

$$dist(CM_4, CM_1) < R = \sqrt{\frac{49area_{pattern}}{27\pi}} \quad (2)$$

A third condition is imposed in the distance between the center of mass of the central ring and that of the main element. Experimentally, it is determined that this distance can not be bigger than 20 pixels, case presented when the pattern is observed at close range under a huge observation angle, from which if a bigger distance is measured during the location process. Thus, the camera focuses on objects alien to our pattern which consequently will be discarded from the search.

Another constrain is introduced to the rate between the area of the main element, $area_4$, and the area of the ellipselike element, $area_4$. This rate has been determined for small angles, and it always lies within the interval [15, 25]. Then, if the calculated rate between the areas of these objects does not lie between the given values, we do not consider those elements part of our pattern.

The fifth and last constrain deals with the area of the ellipselike element, $area_1$, and that of the hollow central ring, $area_0$. Again, from basic mathematics, it can be shown that

$$\frac{area_{disk}}{area_{disk}} = 2. \quad (3)$$

This relation depends on the point of view, but it does not diverge much under small angles. Hence, to avoid being absolutist and battle with possible oscillations of the measurements, we restrict the value of the mentioned relation to values within the interval [1.5, 2.5]¹.

¹ Videos showing the algorithm at work can be watched in the following links:

<http://www.youtube.com/watch?v=6up0LBCXDt0>

<http://www.youtube.com/watch?v=NnRikxv9A8Q>

5 Practical Application

5.1 The Vehicle

To test the applications of our system, we use a small radio controlled car (figure 6) in which we attach a copy of our figure-pattern. We want to drive the vehicle with the only assistance of the camera, that will be pointed to the car, taking images of the scene that will be then processed by an external computer, which is also in charge of producing the control signals and send them through radio to the car.



Fig. 6. The vehicle

5.2 Filtering the Data

Like any other sensor, information gathered by the camera is subject to certain grade of errors. To overcome the slightly deviations of the measurements, a simple and first order fast filter is applied to the estimation of angles and distance. This filter takes the actual measurement and the previous one, weighting each one by a constant factor, so that if a huge discrepancy is found between them, this would be smoothed, avoiding abrupt changes in the control signals. Then, for the angles and the distance, the filter applied is:

$$value_{new} = (1 - K_{filter})value_{old} + K_{filter}value_{measured} \quad (4)$$

In figure 7, the effect of the filter is visible. It shows the evolution taken by the data when K_{filter} takes the values 0.1, 0.4, 0.7 and 0.9. It clearly shows how it smooths the values of the measurements taken by the camera. 0. Similar results are obtained for the distance. We use a value of $K_{filter} = 0.7$ in our tests.

5.3 Car Control

From the state estimation, it is expected to drive a vehicle with the only information obtained by the vision algorithm. As a first approach, we use the distance between the camera and the pattern to decide whether the car moves either forward or backward. Bearing in mind that our goal is testing the usefulness of the vision system, a simple control law is implemented which makes the vehicle moves at slow constant speed in both senses. The distance to the vehicle is divided into three zones: when the vehicle is closer than a certain threshold distance, this moves forward; when it is farther than another threshold distance,

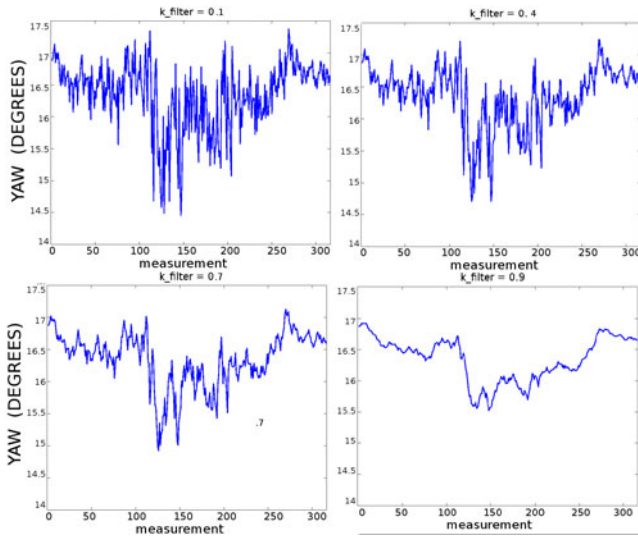


Fig. 7. Evolution of yaw with the value of the filter constant

it moves backward; there is a zone between both thresholds where the vehicle stops. Besides, the estimation of the yaw angle is used to steer the vehicle. For this variable, we use a proportional control, saturating the signal to fit it to the actuator limitations (the servo motor driving our small RC car can steer to angles up to 30 degrees).

5.4 The Hardware

Our system would be uncompleted without some electronics in charge of the vehicle control. The microcontroller Atmega 328 has been chosen, which come integrated in a arduino board ([12] offers extensive and useful information on the use of such development platform, as well as numerous practical examples). This development board is known for for its programming easiness, independently of the operating system used. Even when the programming is performed in C, there are plenty of implemented libraries that make easier the use of external sensors, both analogue and digital. They are developed under the open source philosophy, from which there is a extant community of users keeping them updated and rendering the solution of problems fast and feasible. The essential accompaniment for a wire-free application is an Xbee radio ([13]). We make a radio connection between the PC and the arduino board mounted on the vehicle, what will allow us to send control signals once the vision algorithm is run on the PC. Figure 8 caricatures the layout used to run our test. On the one hand, a printing of the pattern is attached to the vehicle, whose motor speed controller and steering servo are connected to the arduino board, that receives control signal by radio. On the other hand, a handheld camera plugged to a external PC points to the pattern on the vehicle. The PC performs the visual analysis following the

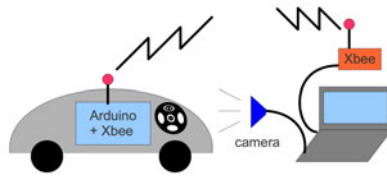


Fig. 8. The experimental layout

described algorithm; with the estimation of the relative position between the camera and the pattern, it sends control signals by radio to the vehicle (the PC is attached with an radio serial adapter [14]).

5.5 The Results

A test is run where a small RC car is driven with the only help of the visual analysis. The pattern is attached to the vehicle and the handheld camera is pointed to it (QuickCam Deluxe from Logitech). The algorithm in test detects the relative position of the camera relative to the pattern, as well as its distance. Measuring the time taken during the capture and analysis of the image and previous values, states change rates are estimated. After analysing the data gathered for the filtering test, we are confident in the usability of our algorithm. The vehicle steers left and right following the movement of the camera. Equally, forward movement seems to be quite smooth due to the speed limitation. We have to remember the speed is kept constant in both sense of directions and stopped in a rank of distances, to make sure we are able to manoeuvre to vehicle within the boundaries of our test room². During the tests, the vehicle is placed in a room illuminated by both natural and artificial light; there is a white, black and grey fleck carpet on the floor room; furniture is present and the vehicle itself offers numerous surfaces that could muddled up the location of our pattern. Besides all these factors, the algorithm is able to extract the pattern out of the frame, estimating the state of the camera relative to it. There are moments when the location is not achieved, probably due to changes in the room light that camouflage the different elements that built up our pattern, and obviously, when the pattern falls out of the field of vision of the camera.

References

1. Pounds, P., Mahony, R., Corke, P.: Modelling and control of a quad-rotor robot. *Aerospace Science and Technology* 11, 183–193 (2007)
2. Bouabdallah, S.: Design and control of quadrotors with application to autonomous flying. PhD Thesis, Ecole Polytechnique Fédérale Lausanne (2007)

² Tests run with the algorithm and vehicle can be watched in the following links:

<http://www.youtube.com/watch?v=uA8xjLpx2Xk>

<http://www.youtube.com/watch?v=AA1Fo3pBjj0>

<http://www.youtube.com/watch?v=6sYSHvmqqRs>

3. Bagen, W., Hu, J., Xu, Y.: School of Aeronautics Science and Engineering, Beijing University, China (2009)
4. Berthold, P.K.: Horn: Robot Vision. McGraw-Hill, New York (1986)
5. Saripalli, S., Montgomery, J.F., Sukhtatme, G.S.: Vision-based landing of an unmanned aerial vehicle. *IEEE Transactions on Robotics and Automation* 19(3), 371–381 (2003)
6. Sharp, C., Shakernia, O., Sastry, S.: A Vision System for Landing an Unmanned Aerial Vehicle. In: *IEEE Int. Conf. on Robotics and Automation* (2001)
7. Cesetti, A., Frontoni, E., Mancini, A., Zingaretti, P.: A Vision-based guidance system for UAV Navigation and safe landing using natural landmarks. *J. Intell Robot Syst.* (57), 233–257 (2009)
8. Dey, S., Bhargab, B., Malay, K., Tinku, A.: A fast algorithm for computing the Euler number of an image and its VTSI implementation. Indian Statistical Institute, Calcuta, India (2000)
9. Xu, G., Zhang, Y., Ji, S., Cheng, Y.: Research on computer vision-based for UAV autonomous landing on a ship. *Pattern Recognition Letters* (30), 600–605 (2009)
10. Lange, S., Snderhauf, N., Protzel, P.: Autonomous Landing for a Multirotor UAV Using Vision. In: *Workshop Proceedings of Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots, SIMPAR 2008, Venice, Italy, November 3-4*, pp. 482–491 (2008)
11. Petruszka, A., Stentz, A.: Stereo vision automatic landing of VTOL UAVS. In: *Proceedings of Assoc. Unmanned Vehicle Syst. Int.*, pages 24563 (1996)
12. Arduino, <http://www.arduino.cc>
13. DIGI International, <http://www.digi.com/products/wireless-wired-embedd>
14. Sparkfun Electronics, <http://www.sparkfun.com>